Discrete Mathematics in Computer Science

E3. Normal Forms and Logical Consequence

Malte Helmert, Gabriele Röger

University of Basel

November 30, 2020

E3.1 Simplified Notation

E3.2 Normal Forms

E3.3 Knowledge Bases

E3.4 Logical Consequences

E3.1 Simplified Notation

Parentheses

Associativity:

$$((\varphi \wedge \psi) \wedge \chi) \equiv (\varphi \wedge (\psi \wedge \chi))$$
$$((\varphi \vee \psi) \vee \chi) \equiv (\varphi \vee (\psi \vee \chi))$$

- Placement of parentheses for a conjunction of conjunctions does not influence whether an interpretation is a model.
- ditto for disjunctions of disjunctions

$\rightarrow$ can omit parentheses and treat this as if parentheses placed arbitrarily

- Example: $(A_1 \wedge A_2 \wedge A_3 \wedge A_4)$ instead of $((A_1 \wedge (A_2 \wedge A_3)) \wedge A_4)$
- Example: $(\neg A \vee (B \wedge C) \vee D)$ instead of $((\neg A \vee (B \wedge C)) \vee D)$

Parentheses

Does this mean we can always omit all parentheses and assume an arbitrary placement? → No!

$$((\varphi \wedge \psi) \vee \chi) \not\equiv (\varphi \wedge (\psi \vee \chi))$$

What should $\varphi \wedge \psi \vee \chi$ mean?

Placement of Parentheses by Convention

Often parentheses can be dropped in specific cases and an implicit placement is assumed:

- $\neg$ binds more strongly than $\wedge$
- $\wedge$ binds more strongly than $\vee$
- $\vee$ binds more strongly than $\rightarrow$ or $\leftrightarrow$

→ cf. PEMDAS/"Punkt vor Strich"

Example
$A \vee \neg C \wedge B \rightarrow A \vee \neg D$ stands for $((A \vee (\neg C \wedge B)) \rightarrow (A \vee \neg D))$

- often harder to read
- error-prone

→ not used in this course

Short Notations for Conjunctions and Disjunctions

Short notation for addition:

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + \cdots + x_n$$
$$\sum_{x \in \{x_1, \ldots, x_n\}} x = x_1 + x_2 + \cdots + x_n$$

Analogously:

$$\bigwedge_{i=1}^{n} \varphi_i = (\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n)$$
$$\bigvee_{i=1}^{n} \varphi_i = (\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n)$$
$$\bigwedge_{\varphi \in X} \varphi = (\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n)$$
$$\bigvee_{\varphi \in X} \varphi = (\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n)$$
$$\text{for } X = \{\varphi_1, \ldots, \varphi_n\}$$

Short Notation: Corner Cases

Is $\mathcal{I} \models \psi$ true for

$$\psi = \bigwedge_{\varphi \in X} \varphi \text{ and } \psi = \bigvee_{\varphi \in X} \varphi$$

if $X = \emptyset$ or $X = \{\chi\}$?

convention:

- $\bigwedge_{\varphi \in \emptyset} \varphi$ is a tautology.
- $\bigvee_{\varphi \in \emptyset} \varphi$ is unsatisfiable.
- $\bigwedge_{\varphi \in \{\chi\}} \varphi = \bigvee_{\varphi \in \{\chi\}} \varphi = \chi$

E3.2 Normal Forms

Why Normal Forms?

- A normal form is a representation with certain syntactic restrictions.
- condition for reasonable normal form: every formula must have a logically equivalent formula in normal form
- advantages:
  - can restrict proofs to formulas in normal form
  - can define algorithms only for formulas in normal form

German: Normalform

Literals, Clauses and Monomials

- A literal is an atomic proposition or the negation of an atomic proposition (e. g., A and $\neg A$).
- A clause is a disjunction of literals (e. g., $(Q \vee \neg P \vee \neg S \vee R)$).
- A monomial is a conjunction of literals (e. g., $(Q \wedge \neg P \wedge \neg S \wedge R)$).

The terms clause and monomial are also used for the corner case with only one literal.

German: Literal, Klausel, Monom

Terminology: Examples

Examples

- $(\neg Q \wedge R)$ is a monomial
- $(P \vee \neg Q)$ is a clause
- $((P \vee \neg Q) \wedge P)$ is neither literal nor clause nor monomial
- $\neg P$ is a literal, a clause and a monomial
- $(P \rightarrow Q)$ is neither literal nor clause nor monomial (but $(\neg P \vee Q)$ is a clause!)
- $(P \vee P)$ is a clause, but not a literal or monomial
- $\neg\neg P$ is neither literal nor clause nor monomial

Conjunctive Normal Form

Definition (Conjunctive Normal Form)

A formula is in conjunctive normal form (CNF)
if it is a conjunction of clauses, i. e., if it has the form

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} L_{ij}$$

with $n, m_i > 0$ (for $1 \leq i \leq n$), where the L_{ij} are literals.

German: konjunktive Normalform (KNF)

Example

$((\neg P \vee Q) \wedge R \wedge (P \vee \neg S))$ is in CNF.

Disjunctive Normal Form

Definition (Disjunctive Normal Form)

A formula is in disjunctive normal form (DNF)
if it is a disjunction of monomials, i. e., if it has the form

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} L_{ij}$$

with $n, m_i > 0$ (for $1 \leq i \leq n$), where the L_{ij} are literals.

German: disjunktive Normalform (DNF)

Example

$((\neg P \wedge Q) \vee R \vee (P \wedge \neg S))$ is in DNF.

CNF and DNF: Examples

Which of the following formulas are in CNF? Which are in DNF?

- $((P \vee \neg Q) \wedge P)$
- $((R \vee Q) \wedge P \wedge (R \vee S))$
- $(P \vee (\neg Q \wedge R))$
- $((P \vee \neg Q) \rightarrow P)$
- P

Construction of CNF (and DNF)

Algorithm to Construct CNF

1. Replace abbreviations $\rightarrow$ and $\leftrightarrow$ by their definitions
   ($(\rightarrow)$-elimination and $(\leftrightarrow)$-elimination).
   $\rightsquigarrow$ formula structure: only $\vee$, $\wedge$, $\neg$
2. Move negations inside using De Morgan and double negation.
   $\rightsquigarrow$ formula structure: only $\vee$, $\wedge$, literals
3. Distribute $\vee$ over $\wedge$ with distributivity
   (strictly speaking also with commutativity).
   $\rightsquigarrow$ formula structure: CNF
4. optionally: Simplify the formula at the end
   or at intermediate steps (e. g., with idempotence).

Note: For DNF, distribute $\wedge$ over $\vee$ instead.

Constructing CNF: Example

Construction of Conjunctive Normal Form

Given: $\varphi = (((P \wedge \neg Q) \vee R) \rightarrow (P \vee \neg(S \vee T)))$

$$
\begin{aligned}
\varphi &\equiv (\neg((P \wedge \neg Q) \vee R) \vee P \vee \neg(S \vee T)) && \text{[Step 1]}\\
&\equiv ((\neg(P \wedge \neg Q) \wedge \neg R) \vee P \vee \neg(S \vee T)) && \text{[Step 2]}\\
&\equiv (((\neg P \vee \neg\neg Q) \wedge \neg R) \vee P \vee \neg(S \vee T)) && \text{[Step 2]}\\
&\equiv (((\neg P \vee Q) \wedge \neg R) \vee P \vee \neg(S \vee T)) && \text{[Step 2]}\\
&\equiv (((\neg P \vee Q) \wedge \neg R) \vee P \vee (\neg S \wedge \neg T)) && \text{[Step 2]}\\
&\equiv ((\neg P \vee Q \vee P \vee (\neg S \wedge \neg T)) \wedge \\
&\quad\ (\neg R \vee P \vee (\neg S \wedge \neg T))) && \text{[Step 3]}\\
&\equiv (\neg R \vee P \vee (\neg S \wedge \neg T)) && \text{[Step 4]}\\
&\equiv ((\neg R \vee P \vee \neg S) \wedge (\neg R \vee P \vee \neg T)) && \text{[Step 3]}
\end{aligned}
$$

Construct DNF: Example

Construction of Disjunctive Normal Form

Given: $\varphi = (((P \wedge \neg Q) \vee R) \rightarrow (P \vee \neg(S \vee T)))$

$$
\begin{aligned}
\varphi &\equiv (\neg((P \wedge \neg Q) \vee R) \vee P \vee \neg(S \vee T)) && \text{[Step 1]}\\
&\equiv ((\neg(P \wedge \neg Q) \wedge \neg R) \vee P \vee \neg(S \vee T)) && \text{[Step 2]}\\
&\equiv (((\neg P \vee \neg\neg Q) \wedge \neg R) \vee P \vee \neg(S \vee T)) && \text{[Step 2]}\\
&\equiv (((\neg P \vee Q) \wedge \neg R) \vee P \vee \neg(S \vee T)) && \text{[Step 2]}\\
&\equiv (((\neg P \vee Q) \wedge \neg R) \vee P \vee (\neg S \wedge \neg T)) && \text{[Step 2]}\\
&\equiv ((\neg P \wedge \neg R) \vee (Q \wedge \neg R) \vee P \vee (\neg S \wedge \neg T)) && \text{[Step 3]}
\end{aligned}
$$

Existence of an Equivalent Formula in Normal Form

Theorem

For every formula φ there is a logically equivalent formula in CNF and a logically equivalent formula in DNF.

- "There is a" always means "there is at least one". Otherwise we would write "there is exactly one".
- Intuition: algorithm to construct normal form works with any given formula and only uses equivalence rewriting.
- actual proof would use induction over structure of formula

Size of Normal Forms

- In the worst case, a logically equivalent formula in CNF or DNF can be exponentially larger than the original formula.
- Example: for $(x_1 \vee y_1) \wedge \cdots \wedge (x_n \vee y_n)$ there is no smaller logically equivalent formula in DNF than:

$$\bigvee_{S \in \mathcal{P}(\{1,\ldots,n\})} \left(\bigwedge_{i \in S} x_i \wedge \bigwedge_{i \in \{1,\ldots,n\} \setminus S} y_i \right)$$

- As a consequence, the construction of the CNF/DNF formula can take exponential time.

More Theorems

Theorem
A formula in CNF is a tautology iff every clause is a tautology.

Theorem
A formula in DNF is satisfiable iff at least one of its monomials is satisfiable.

$\rightsquigarrow$ both proved easily with semantics of propositional logic

E3.3 Knowledge Bases

Knowledge Bases: Example

If not DrinkBeer, then EatFish.
If EatFish and DrinkBeer,
then not EatIceCream.
If EatIceCream or not DrinkBeer,
then not EatFish.

$$\begin{aligned}\text{KB} = \{&(\neg\text{DrinkBeer} \rightarrow \text{EatFish}),\\ &((\text{EatFish} \wedge \text{DrinkBeer}) \rightarrow \neg\text{EatIceCream}),\\ &((\text{EatIceCream} \vee \neg\text{DrinkBeer}) \rightarrow \neg\text{EatFish})\}\end{aligned}$$

Exercise from U. Schöning: Logik für Informatiker
Picture courtesy of graur razvan ionut / FreeDigitalPhotos.net

Models for Sets of Formulas

Definition (Model for Knowledge Base)
Let KB be a knowledge base over A,
i. e., a set of propositional formulas over A.

A truth assignment $\mathcal{I}$ for A is a model for KB (written: $\mathcal{I} \models \text{KB}$)
if $\mathcal{I}$ is a model for every formula $\varphi \in \text{KB}$.

German: Wissensbasis, Modell

Properties of Sets of Formulas

A knowledge base KB is

- satisfiable if KB has at least one model
- unsatisfiable if KB is not satisfiable
- valid (or a tautology) if every interpretation is a model for KB
- falsifiable if KB is no tautology

German: erfüllbar, unerfüllbar, gültig, gültig/eine Tautologie, falsifizierbar

Example I

Which of the properties does $\mathrm{KB} = \{(\mathrm{A} \wedge \neg \mathrm{B}), \neg(\mathrm{B} \vee \mathrm{A})\}$ have?

KB is unsatisfiable:
For every model $\mathcal{I}$ with $\mathcal{I} \models (\mathrm{A} \wedge \neg \mathrm{B})$ we have $\mathcal{I}(\mathrm{A}) = 1$.
This means $\mathcal{I} \models (\mathrm{B} \vee \mathrm{A})$ and thus $\mathcal{I} \not\models \neg(\mathrm{B} \vee \mathrm{A})$.

This directly implies that KB is falsifiable, not satisfiable and no tautology.

Example II

Which of the properties does

$$\begin{aligned}\mathrm{KB} = \{&(\neg \mathrm{DrinkBeer} \rightarrow \mathrm{EatFish}),\\ &((\mathrm{EatFish} \wedge \mathrm{DrinkBeer}) \rightarrow \neg \mathrm{EatIceCream}),\\ &((\mathrm{EatIceCream} \vee \neg \mathrm{DrinkBeer}) \rightarrow \neg \mathrm{EatFish})\} \text{ have?}\end{aligned}$$

- satisfiable, e. g. with $\mathcal{I} = \{\mathrm{EatFish} \mapsto 1, \mathrm{DrinkBeer} \mapsto 1, \mathrm{EatIceCream} \mapsto 0\}$
- thus not unsatisfiable
- falsifiable, e. g. with $\mathcal{I} = \{\mathrm{EatFish} \mapsto 0, \mathrm{DrinkBeer} \mapsto 0, \mathrm{EatIceCream} \mapsto 1\}$
- thus not valid

E3.4 Logical Consequences

Logical Consequences: Motivation

What's the secret of your long life?

I am on a strict diet: If I don't drink beer to a meal, then I always eat fish. Whenever I have fish and beer with the same meal, I abstain from ice cream. When I eat ice cream or don't drink beer, then I never touch fish.

Claim: the woman drinks beer to every meal.

How can we prove this?

Exercise from U. Schöning: Logik für Informatiker
Picture courtesy of graur razvan ionut/FreeDigitalPhotos.net

Logical Consequences

Definition (Logical Consequence)
Let KB be a set of formulas and φ a formula.

We say that KB logically implies φ (written as $\mathrm{KB} \models \varphi$) if all models of KB are also models of φ.

also: KB logically entails φ, φ logically follows from KB, φ is a logical consequence of KB

German: KB impliziert φ logisch, φ folgt logisch aus KB, φ ist logische Konsequenz von KB

Attention: the symbol $\models$ is "overloaded": $\mathrm{KB} \models \varphi$ vs. $\mathcal{I} \models \varphi$.

What if KB is unsatisfiable or the empty set?

Logical Consequences: Example

Let $\varphi = \mathrm{DrinkBeer}$ and

$$\begin{aligned}\mathrm{KB} = \{&(\neg\mathrm{DrinkBeer} \rightarrow \mathrm{EatFish}),\\ &((\mathrm{EatFish} \wedge \mathrm{DrinkBeer}) \rightarrow \neg\mathrm{EatIceCream}),\\ &((\mathrm{EatIceCream} \vee \neg\mathrm{DrinkBeer}) \rightarrow \neg\mathrm{EatFish})\}.\end{aligned}$$

Show: $\mathrm{KB} \models \varphi$

Proof sketch.
Proof by contradiction: assume $\mathcal{I} \models \mathrm{KB}$, but $\mathcal{I} \not\models \mathrm{DrinkBeer}$.
Then it follows that $\mathcal{I} \models \neg\mathrm{DrinkBeer}$.
Because $\mathcal{I}$ is a model of KB, we also have
$\mathcal{I} \models (\neg\mathrm{DrinkBeer} \rightarrow \mathrm{EatFish})$ and thus $\mathcal{I} \models \mathrm{EatFish}$. (Why?)
With an analogous argumentation starting from
$\mathcal{I} \models ((\mathrm{EatIceCream} \vee \neg\mathrm{DrinkBeer}) \rightarrow \neg\mathrm{EatFish})$
we get $\mathcal{I} \models \neg\mathrm{EatFish}$ and thus $\mathcal{I} \not\models \mathrm{EatFish}$. $\rightsquigarrow$ Contradiction!

Important Theorems about Logical Consequences

Theorem (Deduction Theorem)
$\mathrm{KB} \cup \{\varphi\} \models \psi$ *iff* $\mathrm{KB} \models (\varphi \rightarrow \psi)$

German: Deduktionssatz

Theorem (Contraposition Theorem)
$\mathrm{KB} \cup \{\varphi\} \models \neg\psi$ *iff* $\mathrm{KB} \cup \{\psi\} \models \neg\varphi$

German: Kontrapositionssatz

Theorem (Contradiction Theorem)
$\mathrm{KB} \cup \{\varphi\}$ *is unsatisfiable iff* $\mathrm{KB} \models \neg\varphi$

German: Widerlegungssatz

(without proof)